

Graph Spectral Properties of Deterministic Finite Automata

Ryoma Sin'ya

Department of Mathematical and Computing Sciences, Tokyo Institute of Technology.
shinya.r.aa@m.titech.ac.jp

Abstract. We prove that a minimal automaton has a minimal adjacency matrix rank and a minimal adjacency matrix nullity using equitable partition (from graph spectra theory) and Nerode partition (from automata theory). This result naturally introduces the notion of matrix rank into a regular language L , the minimal adjacency matrix rank of a deterministic automaton that recognises L .

We then define and focus on rank-one languages: the class of languages for which the rank of minimal automaton is one. We also define the expanded canonical automaton of a rank-one language.

1 Introduction

The *counting function*¹ $C_L : \mathbb{N} \rightarrow \mathbb{N}$ of a language L over a finite alphabet maps a natural number n into the number of words in L of length n defined as:

$$C_L(n) := |\{w \in L \mid |w| = n\}|.$$

The counting function is a fundamental object in formal language theory and has been studied extensively (*cf.* [1,2]) If L is a regular language, we can represent its counting function $C_L(n)$ using the n -th power of an adjacency matrix of a deterministic automaton that recognises L . Our interest is in the “*easily countable*” class of languages, in the intuitive sense of the word. In this paper, we define and focus on *rank-one languages*: the class of languages that can be recognised by a deterministic automaton for which the adjacency matrix rank is one.

Counting and its applications For any regular language L , it is a well-known result that the counting function of L satisfies:

$$C_L(n) = IM^n F \tag{1.1}$$

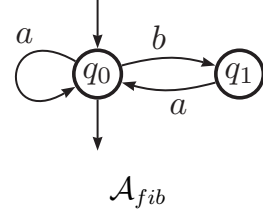
where M is an adjacency matrix, I is an initial vector and F is a final vector of any deterministic automaton recognises L , since $(M^n)_{ij}$ equals to the number of paths of length n from i to j and this corresponds the number of words of length n (*cf.* Lemma 1 in [3]). We give the simple example of Equation (1.1) as follows.

¹ Also called as *growth function*, *generating function* or *combinatorial function*

Example 1. Let \mathcal{A}_{fib} is a deterministic automaton recognises $L = (a + ba)^*$ and

$$M = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \quad I = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad F = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

are its adjacency matrix, initial and final vector. Then the following holds.

$$\begin{aligned} C_L(0) &= |\{\varepsilon\}| = 1, & C_L(1) &= |\{a\}| = 1, \\ C_L(2) &= |\{aa, ba\}| = 2, \\ &\vdots \\ C_L(n) &= IM^n F = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{aligned} \quad (1.2)$$


Equation (1.2) means that $C_L(n)$ equals to the $(n + 1)$ -th Fibonacci number. \diamond

Ranking is one of the variants of counting. The ranking function of L over a finite alphabet A is a bijective function $R_L : L \rightarrow \mathbb{N}$ that maps a word w in L to its index in the lexicographic ordering \prec over A^* defined as:

$$R_L(w) := |\{v \in L \mid v \prec w\}|.$$

In 1985, Goldberg and Sipser introduced a ranking-based string compression in [4]. Recently, the author studied a ranking-based compression on a regular language to analyse its compression ratio and improve a ranking algorithm in [3]. We show an example of a ranking-based compression on a regular language.

Example 2. The formal grammar of *Uniform Resource Identifier* (URI) is defined in RFC 3986 [5], and it is known that the formal grammar of URI is regular (*cf.* [6]). Because the language of all URIs U is regular, we can apply a ranking-based compression on a regular language. For example, the index of the URI $w_1 = \text{http://dlt2014.sciencesconf.org/}$ is:

$$R_U(w_1) = 728552296504796066382113700758455910393907656035063493.$$

The word w_1 is 32 bytes ($|w_1| = 32$), whereas its index $R_U(w_1)$ is 23 bytes ($\lfloor \log_{256} R_U(w_1) \rfloor = 23$). w_1 is compressed up to 72% and, clearly, we can decompress it by the inverse of R_U since ranking is bijective. \diamond

In the case of a ranking on a regular language, the ranking function and its inverse (*unranking*) of L can be calculated using the adjacency matrix of the deterministic automaton L (*cf.* [7,8,3]). Indeed, Example 2 uses RANS[6], which is open source software implemented by the author based on the algorithms in [3].

The computational complexity of an unranking function is higher than a ranking function because the former requires matrix multiplication but the latter does not (*cf.* Table 1 in [3]). In Example 2, the calculation of ranking (compression) was performed in less than one second; however, the calculation of

unranking (decompression) took about two minutes. The reason for such results is that the cost of matrix multiplication is high (the naive algorithm has cubic complexity), and the unranking algorithm requires matrix multiplications, while the ranking algorithm does not. The minimal automaton of U used in Example 2 has 180 states, and its adjacency matrix multiplication cost is high in practice.

Rank-one languages and our results There are several classes of matrices that have a matrix power that can be computed efficiently (*e.g.* diagonalisable matrices and low-rank matrices). We focus on *rank-one matrices* from these classes. As we describe in Section 4, the power of a rank-one matrix has constant time complexity with linear-time preprocessing. We investigate *rank-one languages*: the class of languages for which the rank of minimal automaton is one. We define an automaton as *rank- n* if its adjacency matrix is rank- n . Next, we introduce the definition of the rank of a language.

Definition 1. A regular language L is *rank- n* if there exists a rank- n deterministic automaton that recognises L , and there does not exist a rank- m deterministic automaton that recognises L for any m less than n . \diamond

However, Definition 1 raises the question of how to find a minimal rank. It is a classical theorem in automata theory that for any regular language L , there is a unique automaton \mathcal{A} that recognises L that has a minimal number of states, and \mathcal{A} is called the *minimal automaton* of L . We intend to refine Definition 1 as the following definition.

Definition 1 (refined). A regular language L is *rank- n* if its minimal automaton is rank- n . \diamond

Nevertheless, to achieve this we have to show that a minimal automaton has the minimal rank for consistency of the above two definitions. Hence in Section 3, we prove the following theorem, which has a more general statement.

Theorem 1. *An automaton \mathcal{A} is minimal if and only if both the rank and the nullity of its adjacency matrix are minimal.* \diamond

Theorem 1 provides a necessary and sufficient condition for the minimality of an automaton and is a purely algebraic characterisation of minimal automata. This theorem is not obvious because, in general, for an automaton \mathcal{A} , the number of states of \mathcal{A} and the rank (nullity) of \mathcal{A} are not related. This is illustrated in Figure 1, where the deterministic automaton \mathcal{B}_1 has three states and its rank is two, whereas \mathcal{C}_1 has four states and its rank is one, which equals the rank of the minimal automaton \mathcal{A}_1 . Therefore, we can-not argue naively that “any minimal automaton has the minimal rank (nullity)” by its minimality of states.

The proof consists of the use of two fundamental tools: *equitable partition* from graph spectra theory and *Nerode partition* from automata theory. We briefly introduce these two partitions in Section 2, then give the proof of Theorem 1 in Section 3. In Section 4, we investigate the properties of rank-one languages and introduce expanded canonical automata. In Section 5 we briefly discuss three topics that are not yet understood or lack maturity.

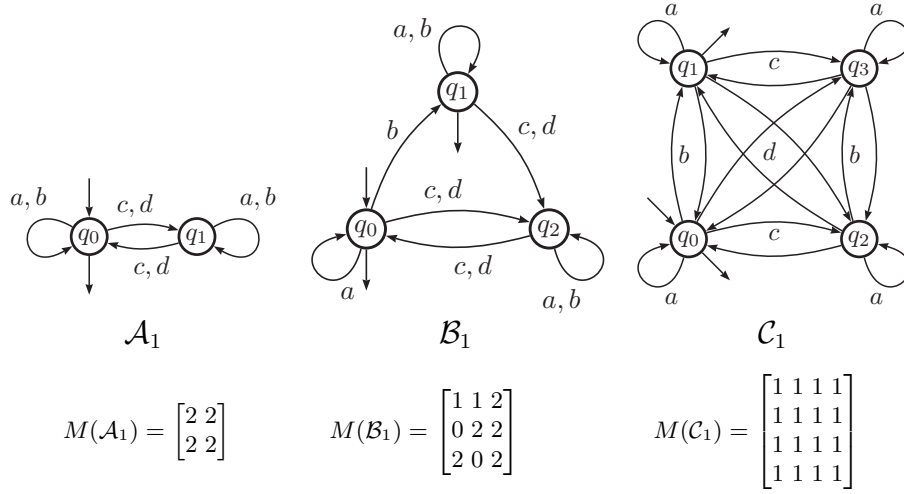


Fig. 1. Three equivalent deterministic trim automata and these adjacency matrices.

2 Nerode partition and equitable partition

We assume that the reader has a basic knowledge of automata, graphs and linear algebra. All results in this section are well-known, and for more details, we refer the reader to [9] for automata theory and [10] for graph spectra theory.

2.1 Automata and languages

A *deterministic finite automaton* \mathcal{A} is a quintuple $\mathcal{A} = \langle Q, A, \delta, q_0, F \rangle$; the finite set of *states* Q , the finite set A called *alphabet*, the *transition function* $\delta : Q \times A \rightarrow Q$, the *initial state* q_0 , and the set of *final states* $F \subseteq Q$. If $\delta(q, w) = p$ is a transition of automaton \mathcal{A} , w is said to be the *label* of the transition. We call a transition $\delta(p, w) = q$ is *successful* if its destination is in the final states F of \mathcal{A} . A word w in A^* is *accepted* by \mathcal{A} if it is the label of a successful transition from the initial state of \mathcal{A} : $\delta(q_0, w) \in F$. The symbol like $|\mathcal{A}|$ denotes the number of states $|Q|$ for an automaton \mathcal{A} . The set of all acceptable words of \mathcal{A} , or *language* of \mathcal{A} , is denoted by $L(\mathcal{A})$. We call two automata are *equivalent* if their languages are identical. An deterministic automaton $\mathcal{A} = \langle Q, A, \delta, q_0, F \rangle$ is *trim* if, for all state $q \in Q$, there exist two words v and w such that $\delta(q_0, v) = q$ (*accessible*) and $\delta(q, w) \in F$ (*co-accessible*).

2.2 Graphs and adjacency matrices

A *multidigraph* G is a pair $G = \langle N, E \rangle$; the set of *nodes* N , the multiset of *edges* E . The *adjacency matrix* $M(G)$ of G is the $|N|$ -dimensional matrix defined as:

$$M(G)_{ij} := \text{the number of edges from node } i \text{ to node } j.$$

The *spectrum* of a matrix M is the multiset of the eigenvalues of M and is denoted by $\lambda(M)$. The *kernel* of a matrix M is the subspace defined as $\{\mathbf{v} \mid M\mathbf{v} = \mathbf{0}\}$, and is denoted $\ker(M)$. We denote the dimension, rank and nullity (the dimension of the kernel) of M by $\dim(M)$, $\text{rank}(M)$ and $\text{null}(M)$ respectively. The following dimension formula is known as the *rank-nullity theorem*:

$$\dim(M) = \text{rank}(M) + \text{null}(M).$$

A *partition* π of a multidigraph $G = \langle N, E \rangle$ is a set of nodal sets $\pi = \{C_1, C_2, \dots, C_k\}$ that satisfies the following three conditions:

$$\emptyset \notin \pi \quad \text{and} \quad \bigcup_{C \in \pi} C = N \quad \text{and} \quad \forall i, j \in [1, k], i \neq j \Rightarrow C_i \cap C_j = \emptyset.$$

We call ${}_{\pi}M$ as the *partitioned matrix* induced by π of M , that is partitioned as

$${}_{\pi}M = \begin{bmatrix} M_{1,1} & \cdots & M_{1,k} \\ \vdots & & \vdots \\ M_{k,1} & \cdots & M_{k,k} \end{bmatrix} \quad (2.1)$$

where the block matrix $M_{i,j}$ is the submatrix of M formed by the rows in C_i and the columns in C_j . The *characteristic matrix* S of π is the $|N| \times k$ matrix that is defined as follows:

$$S_{ij} = \begin{cases} 1 & \text{if } i \in C_j, \\ 0 & \text{otherwise.} \end{cases}$$

In general, S is a full rank matrix ($\text{rank}(S) = \min(|N|, k) = k$) and $S^T S = \text{diag}(|C_1|, |C_2|, \dots, |C_k|)$ where S^T is the transpose of S . The *quotient matrix* M^{π} of M by π is defined as the $k \times k$ matrix:

$$M^{\pi} = (S^T S)^{-1} S^T M S \quad (2.2)$$

where $(S^T S)^{-1} = \text{diag}(\frac{1}{|C_1|}, \frac{1}{|C_2|}, \dots, \frac{1}{|C_k|})$. That is, $(M^{\pi})_{ij}$ denotes the average row sum of the block matrix $({}_{\pi}M)_{i,j}$, in the intuitive sense of the word.

Example 3. Consider the deterministic automaton \mathcal{B}_1 in Figure 1. Let π_1 be the partition of \mathcal{B}_1 : $\pi_1 = \{\{q_0, q_1\}, \{q_2\}\}$, then its characteristic matrix S and $S^T S$, the partitioned matrix ${}_{\pi_1}M(\mathcal{B}_1)$ and the quotient matrix $M(\mathcal{B}_1)^{\pi_1}$ are follows:

$$S = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad S^T S = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \quad {}_{\pi_1}M(\mathcal{B}_1) = \begin{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix} & \begin{bmatrix} 2 \\ 2 \end{bmatrix} \\ \begin{bmatrix} 2 & 0 \end{bmatrix} & \begin{bmatrix} 2 \end{bmatrix} \end{bmatrix}, \quad M(\mathcal{B}_1)^{\pi_1} = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}. \quad \diamond$$

2.3 Nerode partition

Because an automaton \mathcal{A} can be regarded as a multidigraph, we can naturally define the adjacency matrix, partitions and these quotient of \mathcal{A} as the same manner. Let q be a state of \mathcal{A} . We denote by $F(q)$ the set of words w that are labels of a successful transition starting from q . It is called the future of the state q . Two states p and q are said to be *Nerode equivalent* if and only if $F(p) = F(q)$. *Nerode partition* is the partition induced by Nerode equivalence.

Nerode's theorem states that states of minimal automaton are blocks of Nerode partition, edges and terminal states are defined accordingly (cf. [11,12]). That is, note that the adjacency matrix of a minimal automaton equals the quotient matrix of the adjacency matrix of an equivalent automaton by its Nerode partition. For example, π_1 in Example 3 is the Nerode partition of \mathcal{B}_1 in Figure 1 and its induced quotient matrix $M(\mathcal{B}_1)^{\pi_1}$ is identical to the adjacency matrix $M(\mathcal{A}_1)$ of the minimal automaton \mathcal{A}_1 in the same figure.

2.4 Equitable partition

If the row sum of each block matrix $M_{i,j}$ in Equation (2.1) induced by π is constant, then the partition π is called *equitable*. In that case the characteristic matrix S of π satisfies the following equation (cf. Article 15 in [10]):

$$MS = SM^\pi \quad (2.3)$$

If \mathbf{v} is an eigenvector of M^π belonging to the eigenvalue λ , then $S\mathbf{v}$ is an eigenvector of M belonging to the same eigenvalue λ . Indeed, left-multiplication of the eigenvalue equation $M^\pi \mathbf{v} = \lambda \mathbf{v}$ by S yields:

$$\lambda S\mathbf{v} = (SM^\pi)\mathbf{v} = (MS)\mathbf{v} = M(S\mathbf{v}).$$

For example, we can verify that π_1 in Example 3 is equitable. We conclude the the following lemma.

Lemma 1. *Let π be an equitable partition of a matrix M and M^π be its induced quotient matrix, then $\lambda(M^\pi) \subseteq \lambda(M)$ holds.* \square

Remark 1. Though many literature of graph spectra theory including [10] treat the notion of equitable partition on simple graphs, the properties of equitable partition including Equation (2.3) and Lemma 1 are also holds on multidigraphs without problems (cf. [13]; Theorem 4.5).

The concept of equitable partition was introduced in [14]. Equitable partition have been considered in the literature also under the name *divisor* and for more information the reader is referred to Chapter 4 in [13], where basic properties of divisor can be found. \diamond

3 Minimal properties of minimal automata

The “if” direction of the Theorem 1 is obvious from the rank-nullity theorem. For proving the “only if” direction, we prove the following two propositions.

1. Quotient by an equitable partition always reduces the dimension, rank and nullity, respectively (Proposition 3.1).
2. Nerode partition is equitable (Proposition 3.2).

Because, as we mentioned in Section 2.3, the adjacency matrix of a minimal automaton equals to the quotient matrix of the adjacency matrix of any equivalent automaton by its Nerode partition.

Proposition 3.1. *Let π be an equitable partition of a matrix M and M^π be its induced quotient matrix, then the following inequalities hold.*

$$\dim(M^\pi) \leq \dim(M), \quad \text{rank}(M^\pi) \leq \text{rank}(M), \quad \text{null}(M^\pi) \leq \text{null}(M).$$

Proof. $\dim(M^\pi) \leq \dim(M)$ is obvious, then we prove the rest two inequalities.

Let \mathbf{v} be a vector in the kernel of M^π and \mathbf{w} be a vector not in the kernel of M^π , then the following equations hold.

$$M\mathbf{S}\mathbf{v} = S(M^\pi\mathbf{v}) = \mathbf{0}, \tag{3.1}$$

$$M\mathbf{S}\mathbf{w} = S(M^\pi\mathbf{w}) \neq \mathbf{0}. \tag{3.2}$$

Equation (3.2) is induced by $M^\pi\mathbf{w} \neq \mathbf{0}$ and $S\mathbf{u} \neq \mathbf{0}$ for any $\mathbf{u} \neq \mathbf{0}$ since S has full rank. Equation (3.1) and (3.2) leads:

$$\mathbf{v} \in \ker(M^\pi) \Rightarrow S\mathbf{v} \in \ker(M) \quad \text{and} \quad \mathbf{w} \notin \ker(M^\pi) \Rightarrow S\mathbf{w} \notin \ker(M).$$

For any linearly independent vectors \mathbf{u} and \mathbf{u}' then $S\mathbf{u}$ and $S\mathbf{u}'$ are also linearly independent since S has full rank. This shows the rest two inequalities. \square

Proposition 3.2. *Nerode partition is equitable.*

Proof. Let $\mathcal{A} = \langle Q, A, \delta, q_0, F \rangle$ be a deterministic automaton and its Nerode partition $\pi = (C_1, C_2, \dots, C_k); C_i \subseteq Q$. We prove by contradiction.

Assume π is not equitable, then there exist C_i and C_j in π and p and q in C_i such that p and q have different number of transition rules into C_j . We assume without loss of generality that the number of transition rules into C_j of p is larger than q 's. Then there exists at least one alphabet a in A such that $\delta(p, a) \in C_j$ and $\delta(q, a) \notin C_j$. Let $p_a = \delta(p, a)$ and $q_a = \delta(q, a)$, then p_a and q_a are not Nerode equivalent since p_a belongs to another partition of q_a 's. Hence $F(p_a) \neq F(q_a)$ holds and either $F(p_a)$ or $F(q_a)$ is not empty. We assume without loss of generality that $F(p_a)$ is not empty. Because $F(p_a) \neq F(q_a)$ and $F(p_a) \neq \emptyset$, there exists w in $F(p_a)$ such that $w \notin F(q_a)$ and then w satisfies:

$$\delta(p, aw) = \delta(p_a, w) \in F \quad \text{and} \quad \delta(q, aw) = \delta(q_a, w) \notin F.$$

This leads that p and q are not Nerode equivalent even though p and q belong to the same Nerode equivalent class C_i . This is contradiction. \square

It is also proved that any deterministic automaton includes the spectrum of its equivalent minimal automaton by Proposition 3.2 and Lemma 1.

4 Rank-one languages and expanded canonical automata

In this section, we focus on rank-one languages and introduce expanded canonical automata. Firstly, we introduce the well-known general properties of rank-one matrices (*cf.* Proposition 1 in [15]).

Property 1 (characterization of a rank one matrix). Let M , $n \geq 2$, be a n -dimensional real matrix of rank one. Then

1. There exists \mathbf{x}, \mathbf{y} vectors in \mathbb{C}^n ; $\mathbf{x}, \mathbf{y} \neq \mathbf{0}$ such that $M = \mathbf{x}\mathbf{y}^T$;
2. M has at most one non-zero eigenvalue with algebraic multiplicity 1;
3. This eigenvalue is $\mathbf{y}^T \mathbf{x}$. \diamond

Property 1 shows that, for any rank-one language L , its counting function can be represented as a monomial: $C_L(n) = \alpha\lambda^n$ for $n > 0$ and natural numbers α and λ . In addition, rank-one matrices have beneficial property that their power can be computable in constant time with linear-time preprocessing. Indeed, for any m -dimensional rank-one matrix M , there exists \mathbf{x}, \mathbf{y} such that $\mathbf{x}\mathbf{y}^T = M$ hence, the following equation holds for $\lambda = \mathbf{y}^T \mathbf{x}$:

$$M^n = (\mathbf{x}\mathbf{y}^T)^n = \mathbf{x}(\mathbf{y}^T \mathbf{x})^{n-1} \mathbf{y}^T = \lambda^{n-1} \mathbf{x}\mathbf{y}^T = \lambda^{n-1} M.$$

This shows that $(M^n)_{ij}$ equals $\lambda^{n-1}(\mathbf{x}\mathbf{y}^T)_{ij}$, and the inner product of \mathbf{x} and \mathbf{y} has linear-time complexity with respect to its dimension m .

4.1 In-vector and out-vector

For any rank-one matrix M , we can construct \mathbf{x} and \mathbf{y} such that $M = \mathbf{x}\mathbf{y}^T$ from the ratio of the number of incoming edges and outgoing edges, respectively.

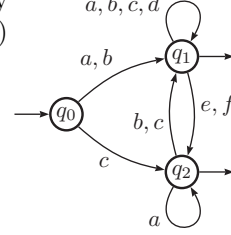
Definition 2. Let M be an n -dimensional rank-one matrix. The *in-vector* of M is a non-zero row vector having minimum length in M and is denoted by $\text{in}(M)$. Because M is rank-one, each row vector \mathbf{v}_i in M can be represented as $\mathbf{v}_i = \alpha_i \cdot \text{in}(M)$ for some natural number $\alpha_i \geq 1$. The *out-vector* of M is the n -dimensional column vector that has an i -th element defined as the above coefficient α_i and denoted by $\text{out}(M)$.

By this construction, it is clear that an in-vector and out-vector satisfy $\text{out}(M) \cdot \text{in}(M) = M$. \diamond

In general, an automaton \mathcal{A} may have the state q such that there are no transition rules into q . Hence, the in-vector must be taken from non-zero vectors in the given matrix. We note that, for any rank-one matrix M , the out-vector of M always contains one because it consists of the coefficients of the in-vector of M (*cf.* Example 4).

Example 4. Consider the rank-one automaton shown in the adjacent figure. This automaton is deterministic and trim. Its adjacency matrix M and in-vector $\text{in}(M)$ and out-vector $\text{out}(M)$ are follows.

$$M = \begin{bmatrix} 0 & 2 & 1 \\ 0 & 4 & 2 \\ 0 & 2 & 1 \end{bmatrix}, \quad \text{in}(M) = [0, 2, 1], \quad \text{out}(M) = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}.$$



Note that $\text{in}(M)_0 = 0$ means that q_0 has no incoming transition rules. \diamond

4.2 Expanded canonical automata

First, we define a normal form of a rank-one automaton.

Definition 3. A rank-one automaton \mathcal{A} is *expanded normal* if for its adjacency matrix M , each element of the in-vector of M equals to zero or one. \diamond

The automaton in Figure 4 is not expanded normal because the second element of its in-vector equals two. Expanded normal form is a graph normal form of automata, and does not consider labels.

Secondly, we propose the operation *expansion* that expands the given matrix (graph or automaton) algebraically.

Definition 4. Let ${}^\pi M$ and M be two matrices of dimension m and n , respectively. We define ${}^\pi M$ as an expansion of M if there exists a partition $\pi = \{C_1, C_2, \dots, C_n\}$ of ${}^\pi M$ such that the characteristic matrix S of π satisfies:

$${}^\pi M = SM(S^T S)^{-1} S^T. \quad \diamond$$

Expansion is an algebraic transformation that increases the dimension of the given matrix. Intuitively, expansion can be regarded as an inverse operation of quotient. Indeed, for any expanded matrix ${}^\pi M$ of some n -dimensional M by π and its characteristic matrix S , we have the following equation:

$$({}^\pi M)^\pi = (S^T S)^{-1} S^T ({}^\pi M) S = (S^T S)^{-1} S^T (SM(S^T S)^{-1} S^T) S = M$$

which holds by Equation (2.2) and Definition 4. If M is rank-one, then for any expanded matrix ${}^\pi M$ of M , the out-vector of ${}^\pi M$ consists of same elements as those of the out-vector of M . This reflects the invariance of the number of outgoing transition rules of the Nerode equivalent states (*cf.* Figure 2).

Finally, we define a canonical automaton of a rank-one language: *expanded canonical automaton*. The minimal automaton of a regular language K is uniquely determined by K , whereas the expanded canonical automaton of a rank-one language L is not uniquely determined, but its graph structure is uniquely determined by L .

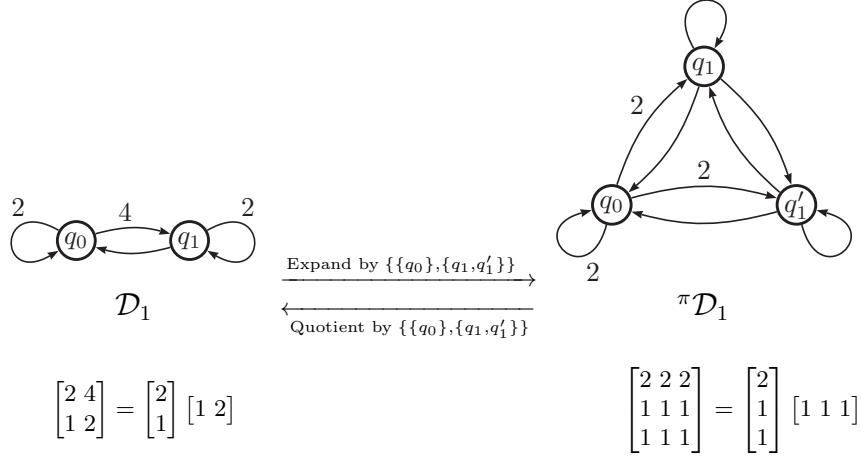


Fig. 2. The rank-one graph \mathcal{D}_1 and its expanded canonical form ${}^\pi \mathcal{D}_1$.

Definition 5. Let L be a rank-one language, then we define its expanded canonical automaton ${}^\pi \mathcal{A}_L$ as the expanded automaton of the minimal automaton \mathcal{A}_L of L by a partition $\pi = \{C_1, C_2, \dots, C_{|\mathcal{A}_L|}\}$ such that, for all $C_i \in \pi$, $|C_i| = \text{in}(\mathcal{A}_L)_i$ if $\text{in}(\mathcal{A}_L)_i \neq 0$ and 1 otherwise. \diamond

By the definition, it is clear that for any rank-one language L , its expanded canonical automaton is expanded normal (*cf.* Figure 2, or \mathcal{A}_1 and its expanded canonical automaton \mathcal{C}_1 in Example 1). As we describe in Section 5.1, we introduce expanded canonical automata for analysis and evaluation of the closure properties of rank-one languages. Because of the limitations of space, a detailed discussion of expanded canonical automata is not possible here.

5 The way for further developments

5.1 Closure property of rank-one languages and decomposability

It is natural to consider the closure properties of rank-one languages. However, with some exceptions (*e.g.* *quotient*, *prefix*²), the class of rank-one languages is, for the most part, not closed under an operation on languages: *e.g.* *union*, *concatenation* and *Kleene star*. Indeed, for the two rank-one (expanded canonical) automaton \mathcal{E}_1 and \mathcal{F}_1 in Figure 3, the union of $L(\mathcal{E}_1)$ and $L(\mathcal{F}_1)$ has the spectrum $\{3, 2, -1, -1, -1\}$ (without zeros) and is rank-five. We note that $L(\mathcal{E}_1)$ and $L(\mathcal{F}_1)$ have the same prefix, and the minimal automaton of $L(\mathcal{E}_1) \cup L(\mathcal{F}_1)$ is strongly connected and has nine states. In addition, note that there exist rank-one languages K and L such that the union language $K \cup L$ has irrational and complex eigenvalues.

² The operations that can be realised without destroying graph structure of a deterministic automaton

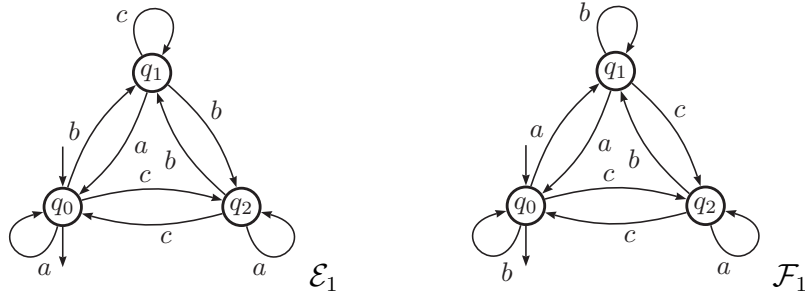


Fig. 3. Two rank-one automata (expanded canonical automata).

Conversely, we consider the closure of rank-one languages with an operation on languages or decomposability into rank-one languages (*rank-one decomposition*). In the case of matrices, matrix rank-one decomposition is well studied and there exist fundamental results such as *orthogonal decomposition* for real symmetric matrices. We are interested in investigating regular language rank-one decomposition.

5.2 Rank of unambiguous automata

The class of *unambiguous automata* is a more general class of automata than the class of deterministic automata (cf. [9]). We intend to generalise Definition 1 as Definition 6 which is more essential for the counting structure of languages because unambiguous automata is the most general class that satisfies Equality (1.1). It will be interesting to determine whether the rank of a minimal unambiguous automaton is minimal. If so, we can refine Definition 6 in a similar manner.

Definition 6. A regular language L is *unambiguous rank- n* if there exists a rank- n unambiguous automaton recognises L and does not exist a rank- m unambiguous automaton recognises L for any m less than n . \diamond

5.3 Relation between the conjugacy of automata

Béal *et al.* developed the theory of *conjugacy of automata* (cf. [16,17]) that gives structural information on two equivalent \mathbb{K} -automata. Conjugacy of automata is a theory based on matrices, and we think some results in this paper may be reconstructed by the theory of conjugacy.

Acknowledgement I would like to thank my adviser, Kazuyuki Shudo, for his continuous support and encouragement. Special thanks also go to Yuuya Uezato, is a postgraduate student at University of Tsukuba, who provided carefully considered feedback and valuable comments.

References

1. Shur, A.M.: Combinatorial complexity of regular languages. In: Proceedings of the 3rd International Conference on Computer Science: Theory and Applications. CSR'08, Berlin, Heidelberg, Springer-Verlag (2008) 289–301
2. Shur, A.M.: Combinatorial characterization of formal languages. CoRR **abs/1010.5456** (2010)
3. Sin'ya, R.: Text compression using abstract numeration system on a regular language. Computer Software **30**(3) (2013) 163–179 in Japanese, English extended abstract is available at <http://arxiv.org/abs/1308.0267>.
4. Goldberg, A., Sipser, M.: Compression and ranking. In: Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing. STOC '85, New York, NY, USA, ACM (1985) 440–448
5. Berners-Lee, T., Fielding, R., Masinter, L.: Rfc 3986, uniform resource identifier (uri): Generic syntax (2005)
6. Sin'ya, R.: Rans : More advanced usage of regular expressions. <http://sinya8282.github.io/RANS/>
7. Choffrut, C., Goldwurm, M.: Rational transductions and complexity of counting problems. Mathematical Systems Theory **28**(5) (1995) 437–450
8. Lecomte, P., Rigo, M.: Combinatorics, Automata and Number Theory. 1st edn. Cambridge University Press, New York, NY, USA (2010) Chapter 3: Abstract numeration systems.
9. Sakarovitch, J.: Elements of Automata Theory. Cambridge University Press, New York, NY, USA (2009)
10. Miegheem, P.V.: Graph Spectra for Complex Networks. Cambridge University Press, New York, NY, USA (2011)
11. Nerode, A.: Linear automaton transformations. Proceedings of the American Mathematical Society **9**(4) (1958) 541–544
12. Béal, M.P., Crochemore, M.: Minimizing incomplete automata. In: Finite-State Methods and Natural Language Processing (FSMNLP'08). Joint Research Center (2008) 9–16
13. Cvetković, D.M., Doob, M., Sachs, H.: Spectra of Graphs: Theory and Application. Academic Press, New York (1980)
14. Schwenk, A.: Computing the characteristic polynomial of a graph. In Bari, R., Harary, F., eds.: Graphs and Combinatorics. Volume 406 of Lecture Notes in Mathematics. Springer Berlin Heidelberg (1974) 153–172
15. Osnaga, S.M.: On rank one matrices and invariant subspaces. Balkan Journal of Geometry and its Applications (BJGA) **10**(1) (2005) 145–148
16. Béal, M.P., Lombardy, S., Sakarovitch, J.: On the equivalence of z-automata. In: Proceedings of the 32Nd International Conference on Automata, Languages and Programming. ICALP'05, Berlin, Heidelberg, Springer-Verlag (2005) 397–409
17. Béal, M.P., Lombardy, S., Sakarovitch, J.: Conjugacy and equivalence of weighted automata and functional transducers. In: Proceedings of the First International Computer Science Conference on Theory and Applications. CSR'06, Berlin, Heidelberg, Springer-Verlag (2006) 58–69